L Number	Hits	Search Text	DB	Time stamp
1	16	maintain$4 with (permanent adj connect$4) with network	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/25 13:40
2	2061	TCP adj connect$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/25 13:35
3	14	(fail$4 adj over or (failed-over)) adj connect$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/25 13:47
4	3	(TCP adj connect$4) and ((fail$4 adj over or (failed-over)) adj connect$4)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/25 13:36
5	2	(maintain$4 with (permanent adj connect$4) with network) and (TCP adj connect$4)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/25 13:40
7	2	(TCP adj connect$4) and (fail-over adj policy)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/25 13:41
8	18729	IP adj address$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/25 13:41
9	2	ARP adj ownership adj policy	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/25 13:42
10	4056	MAC adj address$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/25 13:43
11	2336	(fail$4 adj over or (failed-over))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/25 13:49

12	231	(TCP adj connect$4) and (MAC adj address$4)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/25 13:49
13	17	((fail$4 adj over or (failed-over))) and ((TCP adj connect$4) and (MAC adj address$4))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/25 13:49
6	6	fail-over adj policy	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/25 13:56
15	4	ownership adj recover$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/25 13:58
16	2	(recover$4 same connect$4 same ((fail$4 adj over or (failed-over)))) and (MAC adj address$4) and (IP adj address$4) and (TCP adj connect$4)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/25 13:58
17	3	(recover$4 same connect$4 same ((fail$4 adj over or (failed-over)))) and (MAC adj address$4) and (IP adj address$4)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/25 13:58
18	2	(recover$4 same connect$4 same ((fail$4 adj over or (failed-over)))) and (MAC adj address$4) and (TCP adj connect$4)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/25 13:59
19	4	(recover$4 same connect$4 same ((fail$4 adj over or (failed-over)))) and (IP adj address$4) and (TCP adj connect$4)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/25 13:59
20	16	(recover$4 same connect$4 same ((fail$4 adj over or (failed-over)))) and (IP adj address$4)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/25 14:00
21	4	(recover$4 same connect$4 same ((fail$4 adj over or (failed-over)))) and (TCP adj connect$4)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/25 14:00

22	3	(recover$4 same connect$4 same ((fail$4 adj over or (failed-over)))) and (MAC adj address$4)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/25 14:01
14	40	recover$4 same connect$4 same ((fail$4 adj over or (failed-over)))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/09/25 14:01

US-PAT-NO: 6477139

DOCUMENT-IDENTIFIER: US 6477139 B1

TITLE: Peer controller management in a dual controller fibre channel storage enclosure

---------- KWIC ---------

Brief Summary Text - BSTX (5):

Because of the high bandwidth and flexible connectivity provided by the FC, the FC is becoming a common medium for interconnecting peripheral devices within multi-peripheral-device enclosures, such as redundant arrays of inexpensive disks ("RAIDs"), and for **connecting** multi-peripheral-device enclosures with one or more host computers. These multi-peripheral-device enclosures economically provide greatly increased storage capacities and built-in redundancy that facilitates mirroring and **fail over** strategies needed in high-availability systems. Although the PC is well-suited for this application with regard to capacity and connectivity, the FC is a serial communications medium. Malfunctioning peripheral devices and enclosures can, in certain cases, degrade or disable communications. A need has therefore been recognized for methods to improve the ability of FC-based multi-peripheral-device enclosures to isolate and **recover** from malfunctioning peripheral devices. A need has also been recognized for additional communications and component redundancies within multi-peripheral-device enclosures to facilitate higher levels of fault-tolerance and high-availability.

US-PAT-NO: 6266781

DOCUMENT-IDENTIFIER: US 6266781 B1
See image for Certificate of Correction

TITLE: Method and apparatus for providing failure detection and recovery with predetermined replication style for distributed applications in a network

---------- KWIC ---------

Brief Summary Text - BSTX (13):

In accordance with the present invention, an application module running on a host computer is made reliable by first registering itself for its own failure and **recovery** processes. A ReplicaManager daemon process, running on the same host computer on which the application module is running or on another host computer **connected** to the network to which the application module's machine is **connected,** receives a registration message from the application module. This registration message, in addition to identifying the registering application module and the host machine on which it is running, includes the particular replication strategy (cold, warm or hot backup style) and the degree of replication to be associated with the registered application module, which registered replication strategy is used by the ReplicaManager to set the operating state of each backup copy of the application module as well as to maintain the number of backup copies in accordance with the degree of replication. A Watchdog daemon process, running on the same host computer as the registered application module then periodically monitors the registered application module to detect failures. When the Watchdog daemon detects a crash or a hangup of the monitored application module, it reports the failure to the ReplicaManager, which in turn effects a **fail-over** process. Accordingly, if the replication style is warm or hot and the failed application module cannot be restarted on its own host computer, one of the running backup copies of the primary application module is designated as the new primary application module and a host computer on which an idle copy of the application

module resides is signaled over the network to execute that idle application. The degree of replication is thus maintained thereby assuring protection against multiple failures of that application module. If the replication style is cold and the failed application is cannot be restarted on its own host computer, then a host computer on which an idle copy of the application module resides is signaled over the network to execute the idle copy. In order to detect a failure of a host computer or the Watchdog daemon running on a host computer, a SuperWatchDog daemon process, running on the same host computer as the ReplicaManager, detects inputs from each host computer. Upon a host computer failure, detected by the SuperWatchDog daemon by the lack of an input from that host computer, the ReplicaManager is accessed to determine the application modules that were running on that host computer. Those application modules are then individually failure-protected in the manner established and stored in the ReplicaManager.

US-PAT-NO: 6260079

DOCUMENT-IDENTIFIER: US 6260079 B1

TITLE: Method and system for enhancing fibre channel loop resiliency for a mass storage enclosure by increasing component redundancy and using shunt elements and intelligent bypass management

---------- KWIC ---------

Brief Summary Text - BSTX (5):

Because of the high bandwidth and flexible connectivity provided by the FC, the FC is becoming a common medium for interconnecting peripheral devices within multi-peripheral-device enclosures, such as redundant arrays of inexpensive disks ("RAIDs"), and for **connecting** multi-peripheral-device enclosures with one or more host computers. These multi-peripheral-device enclosures economically provide greatly increased storage capacities and built-in redundancy that facilitates mirroring and **fail over** strategies needed in high-availability systems. Although the FC is well-suited for this application with regard to capacity and connectivity, the FC is a serial communications medium. Malfunctioning peripheral devices and enclosures can, in certain cases, degrade or disable communications. A need has therefore been recognized for methods to improve the ability of FC-based multi-peripheral-device enclosures to isolate and **recover** from malfunctioning peripheral devices, and for improving the ability of systems including one or more host computers and multiple, interconnected FC-based multi-peripheral-device enclosures to isolate and **recover** from a malfunctioning multi-peripheral-device enclosure. A need has also been recognized for additional communications and component redundancies within multi-peripheral-device enclosures to facilitate higher levels of fault-tolerance and high-availability.

US-PAT-NO: 6195760

DOCUMENT-IDENTIFIER: US 6195760 B1

TITLE: Method and apparatus for providing failure detection and recovery with predetermined degree of replication for distributed applications in a network

---------- KWIC ---------

Brief Summary Text - BSTX (13):

In accordance with the present invention, an application module running on a host computer is made reliable by first registering itself for its own failure and **recovery** processes. A ReplicaManager daemon process, running on the same host computer on which the application module is running or on another host computer **connected** to the network to which the application module's machine is **connected,** receives a registration message from the application module. This registration message, in addition to identifying the registering application module and the host machine on which it is running, includes the particular replication strategy (cold, warm or hot backup style) and the degree of replication to be associated with the registered application module, which registered replication strategy is used by the ReplicaManager to set the operating state of each backup copy of the application module as well as to maintain the number of backup copies in accordance with the degree of replication. A Watchdog daemon process, running on the same host computer as the registered application module then periodically monitors the registered application module to detect failures. When the Watchdog daemon detects a crash or a hangup of the monitored application module, it reports the failure to the ReplicaManager, which in turn effects a **fail-over** process. Accordingly, if the replication style is warm or hot and the failed application module cannot be restarted on its own host computer, one of the running backup copies of the primary application module is designated as the new primary application module and a host computer on which an idle copy of the application

module resides is signaled over the network to execute that idle application. The degree of replication is thus maintained thereby assuring protection against multiple failures of that application module. If the replication style is cold and the failed application is cannot be restarted on its own host computer, then a host computer on which an idle copy of the application module resides is signaled over the network to execute the idle copy. In order to detect a failure of a host computer or the Watchdog daemon running on a host computer, a SuperWatchDog daemon process, running on the same host computer as the ReplicaManager, detects inputs from each host computer. Upon a host computer failure, detected by the SuperWatchDog daemon by the lack of an input from that host computer, the ReplicaManager is accessed to determine the application modules that were running on that host computer. Those application modules are then individually failure-protected in the manner established and stored in the ReplicaManager.

US-PAT-NO: 6185601

DOCUMENT-IDENTIFIER: US 6185601 B1

TITLE: Dynamic load balancing of a network of client and server computers

---------- KWIC ---------

Detailed Description Text - DETX (70):

FIG. 4B shows which of the software modules, described and discussed above in correction with FIG. 2B, is associated with the processing by an aware client of a **fail-over** or fail-back on the network. **Fail-over** refers to the response, by aware clients seeking access to a resource, to the failure of a node, e.g. server, designated in the name driver module 194 for accessing that resource. Fail-back deals with the behavior of an aware client in response to a **recovery** of a node, e.g. server, on the network from a failed condition. The operation begins, in a manner similar to that described and discussed above in **connection** with FIG. 4A, with the issuance of an I/O request by the application module 196. That request is passed to the command processing module 192. Since the I/O request is destined for an external resource, the path to the resource needs to be determined. The request is therefore passed to the resource management module 186 and to the name driver module 194 to obtain the path. The command processing module 192 passes the request with path information to **fail-over** module 188 for further processing. **Fail-over** module 188 then calls the redirector module 184 to send the I/O request via the path obtained from the name driver. If **fail-over** module 188 determines that there is a failure, it calls the name driver module to provide an alternate path for the I/O operation, and the **fail-over** module 188 reissues the I/O command with the alternate path to the redirector module 184. Data passing between the resource and the application module 196 is passed via the redirector module 184. Upon failure detection and redirecting by **fail-over** module 188, name

driver module 194 marks the path as failed. Periodically, name driver module 194 checks the network for the valid presence of the failed paths and, if good, once again marks them failed-back or valid so that they may once again be used in the future, if necessary.

US-PAT-NO: 6101508

DOCUMENT-IDENTIFIER: US 6101508 A

TITLE: Clustered file management for network resources

---------- KWIC ---------

Detailed Description Text - DETX (73):
FIG. 4B shows which of the software modules, described and discussed above in **connection** with FIG. 2B, is associated with the processing by an aware client of a **fail-over** or fail-back on the network. **Fail-over** refers to the response, by aware clients seeking access to a resource, to the failure of a node, e.g. server, designated in the name driver module 194 for accessing that resource. Fail-back deals with the behavior of an aware client in response to a **recovery** of a node, e.g. server, on the network from a failed condition. The operation begins, in a manner similar to that described and discussed above in **connection** with FIG. 4A, with the issuance of an I/O request by the application module 196. That request is passed to the command processing module 192. Since the I/O request is destined for all external resource, the path to the resource needs to be determined. The request is therefore passed to the resource management module 186 and to the name driver module 194 to obtain the path. The command processing module 192 passes the request with path information to **fail-over** module 188 for further processing. **Fail-over** module 188 then calls the redirector module 184 to send the I/O request via the path obtained from the name driver. If **fail-over** module 188 determines that there is a failure, it calls the name driver module to provide an alternate path for the I/O operation, and the **fail-over** module 188 reissues the I/O command with the alternate path to the redirector module 184. Data passing between the resource and the application module 196 is passed via the redirector module 184. Upon failure detection and redirecting by **fail-over** module 188, name driver module 194 marks the path as failed. Periodically, name driver module

194 checks the network for the valid presence of the failed paths and, if good, once again marks them failed-back or valid so that they may once again be used in the future, if necessary.

US-PAT-NO: 6067545

DOCUMENT-IDENTIFIER: US 6067545 A

TITLE: Resource rebalancing in networked computer systems

---------- KWIC ---------

Detailed Description Text - DETX (102):

FIG. 4B shows which of the software modules described and discussed above in **connection** with FIG. 2B is associated with the processing by an aware client of a **fail-over** or fail-back on the network. **Fail-over** refers to the response by aware clients seeking access to a resource to the failure of a node, e.g. server, designated in the name driver module 194 for accessing that resource. Fail-back deals with the behavior of an aware client in response to a **recovery** of a node, e.g. server, on the network from a failed condition. The operation begins in a manner similar to that described and discussed above in **connection** with FIG. 4A with the issuance of an I/O request by the application module 196. That request is passed to the command processing module 192. Since the I/O request is destined for an external resources the path to the resource needs to be determined. The request is therefore passed to the resource management module 186 and to the name driver module 194 to obtain the path. The command processing module 192 passes the request with path information to **fail-over** module 188 for further processing. **Fail-over** module 188 then calls the redirector module 184 to send the I/O request via the path obtained from the name driver. If **fail-over** module 188 determines there is a failure it calls the name driver module to provide an alternate path for the I/O operation and the **fail-over** module 188 reissues the I/O command with the alternate path to the redirector module 184. Data passing between the resource and the application module 196 is passed via the redirector module 184. Upon failure detection and redirecting by **fail-over** module 188, name driver module 194 marks the path as failed. Periodically name driver module 194 checks the network for

the valid presence of the failed paths and if good, once again marks them
failed-back or valid so that they may once again be used in the future if
necessary.